12112

NETWORK WORKING GROUP
Request for Comments #407
NIC #12112
Obsoletes: RFC #360

Bob Bressler (MIT-DMCG)
Rich Guida (MIT-DMCG)
Alex McKenzie (BBN-NET)

REMOTE JOB ENTRY PROTOCOL

By delegation of the ARPA office, the attached document is released as the official RJE Protocol. This document is an extension and clarification of RFC 360, written by Chuck Holland. It incorporates some of the ideas presented by Bob Braden in RFC 368. It also conforms to RFC 354 (as modified by RFC 385), the File Transfer Protocol by Abhay Bhushan, insofar as syntactic definitions are concerned.

A user-package exists on the MIT-DMCG PDP-10, and has been somewhat debugged locally, but not **over** the ARPANET, naturally, since there are no servers yet. However, it is hoped that the existence of this user-package will serve as some sort of incentive for servers to implement the protocol as soon as possible; as soon as one is ready, just contact Bob Bressler or Rich Guida and testing can begin immediately.

Network RJE

~~Questions: what subset of FTP is required?~~

Comments on RFC 407

① Telnet break / synch { ? cancel / delete character
② Message facility?
③ Can't look up usercode + password together — sequence when omit PASS.
④ How change RJE password?
⑤ p. 8 - Make decimal integer the default
⑥ p. 8 - colon in <attributes> unnecessary (use , ?)
⑦ p. 8 - why can't omit host from <host file>
⑧ Telnet - how about command completion?
⑨ ~~How about~~ Why restrict data byte size to 8 bits?

~~Questions/Suggestions~~ for CCN:

(1) What subset of FTP required?
(2) How save output? Dupe jobname?
(3) Output routing info with job.
(4) Userid conventions

Comments / Questions on ~~#~~ RFC #407

P.5 1.

This implies that an NVT ~~should~~, allows arbitrary input line length, Is this true? Consistent with reality?
~~Is a continuation convention desirable~~ Thus, the RJE protocol implies that
~~.~~ any continuation convention required when a single command exceed one physical line on the user's terminal should be provided in the user Telnet rather than in the sever RJE protocol.

(Oct. 16, 1972)
RFC 407 NIC 12112

Obsoletes RFC 360

bert Bressler, MIT-DMCG
chard Guida, MIT-DMCG
Alex McKenzie, BBN-NET

REMOTE JOB ENTRY PROTOCOL

REMOTE JOB ENTRY PROTOCOL

INTRODUCTION

Remote job entry is the mechanism whereby a user at one location causes a batch-processing job to be run at some other location. This protocol specifies the Network standard procedures for such a user to communicate over the Network with a remote batch-processing server, causing that server to retrieve a job-input file, process the job, and deliver the job's output file(s) to a remote location. The protocol uses a TELNET connection (to a special standardized logger, not socket 1) for all control communication between the user and the server RJE processes. The server-site then uses the File Transfer Protocol to retrieve the job-input file and to deliver the output file(s).

There are two types of users: direct users (persons) and user processes. The direct user communicates from an interactive terminal attached to a TIP or any host. This user may cause the input and/or output to be retrieved/sent on a specific socket at the specified host (such as for card readers or printers on a TIP), or the user may have the files transferred by file-id using File Transfer Protocol. The other type of user is a RJE User-process in one remote host communicating with the RJE Server-process in another host. This type of user ultimately receives its instructions from a human user, but through some unspecified indirect means. The command and response streams of this protocol are designed to be readily used and interpreted by both the human user and the user process.

A particular user site may choose to establish the TELNET control connection for each logical job or may leave the control connection open for extended periods. If the control connection is left open, then multiple job-files may be directed to be retrieved or optionally (to servers that are able to determine the end of one logical job by the input stream and form several jobs out of one input file) one continuous retrieval may be done (as from a TIP card reader). This then forms a "hot" card reader to a particular server with the TELNET connection serving as a "job monitor". Since the output is always transferred job at a time per connection to the output socket, the output from this "hot" reader would appear when ready as if to a "hot" printer. Another possibility for more complex hosts is to attach an RJE User-process to a card reader and take instructions from a lead control card, causing an RJE control TELNET to be opened to the appropriate host with appropriate log-on and input retrieval commands. This card reader would appear to the human user as a Network "hot" card reader. The details of this RJE User-process are beyond the scope of this protocol.

GENERAL SPECIFICATIONS

User

A human user at a real terminal or a process that supplies the command control stream causing a job to be submitted remotely will be termed the User. The procedure by which a process user receives its instructions is beyond the scope of this protocol.

User TELNET

The User communicates its commands over the Network in Network Virtual Terminal code through a User TELNET process in the User's Host. This User TELNET process initiates its activity via ICP to the standard "RJE Logger" socket (socket 5) at the desired RJE-server Host.

RJE-Server TELNET

The RJE-server process receives its command stream from and sends its response stream to the TELNET channel through an RJE-server TELNET process in the server host. This process must listen for the ICP on the "RJE Logger" socket (and cause appropriate ICP socket shifting).

TELNET Connection

The command and response streams for the RJE mechanism are via a TELNET-like connection to a special socket with full specifications according to the current NWG TELNET protocol.

RJE-Server

The RJE-Server process resides in the Host which is providing Remote Batch Job Entry service. This process receives input from the RJE-server TELNET, controls access through the "log-on" procedure, retrieves input job files, queues jobs for execution by the batch system, responds to status inquiries, and transmits job output files when available.

User FTP

All input and output files are transferred under control of the RJE-server process at its initiative. These files may be directly transferred via Request-for-connection to a specific Host/socket or they may be transferred via File Transfer Protocol. If the latter method is used, then the RJE-server acts through its local User FTP process to cause the transfer. This process initiates

activity by an active Request-for-connection to the "FTP Logger" in the foreign host.

Server FTP

This process in a remote host (remote from the RJE-server) listens for an ICP from the User FTP and then acts upon the commands from the User FTP causing the appropriate file transfer.

FTP

When File Transfer Protocol is used for RJE files, the standard FTP mechanism is used as fully specified by the current NWG FTProtocol.

RJE Command Language

The RJE system is controlled by a command stream from the User over the TELNET connection specifying the user's identity (log-on), the source of the job input file, the disposition of the job's output files, enquiring about job status, altering job status or output disposition. Additional commands affecting output disposition are includable in the job input file. This command language is explicitly specified in a following section of this protocol.

RJE Command Replies

Every command input from the User via TELNET calls for a response message from the RJE-server to the User over the TELNET connection. Certain other conditions also require a response message. These messages are formatted in a standardized manner to facilitate interpretation by both human Users and User processes. A following section of this protocol specifies the response messages.

RJE COMMANDS OVER TELNET CONNECTION

GENERAL CONVENTIONS

1. Each of the commands will be contained in one input line terminated by the standard TELNET "crlf". The line may be of any length desired by the user (explicitly, not restricted to a physical terminal line width). The characters "cr" and "lf" will be ignored by the RJE-server except in the explicit order "crlf" and may be used as needed for local terminal control.

2. All commands will begin with a recognized command name and may then contain recognized syntactic element strings and free-form variable strings (for user-id, file-ids, etc.). Recognized words consist of alphanumeric strings (letters and digits) or punctuation. Recognized alphanumeric string elements must be separated from each other and from unrecognizable strings by at least one blank or a syntacticly permitted punctuation. Other blanks may be used freely as desired before or after any syntactic element ("blank" is understood here to mean ASCII SPACE (octal 040); formally: <blank>::= <blank><ASCII SPACE> | <ASCII SPACE> ; thus, a sequence of SPACES is also permissible in place of <blank>, although there is no syntactic necessity for there to be more than one). The "=" after the command name in all commands except OUT and CHANGE is optional.

3. Recognized alphanumeric strings may contain upper case letters or lower case letters in any mixture without syntactic differentiation. Unrecognizable strings will be used exactly as presented with full differentiation of upper and lower case input, unless the host finally using the string defines otherwise.

4. There are two types of Unrecognizable strings: final and imbedded. Final strings appear as the last syntactic element of a command and are parsed as beginning with the next non-blank character of the input stream and continuing to the last non-blank character before the "crlf".

Imbedded strings include "job-id" and "job-file-id" in the OUT, CHANGE, and ALTER commands. At present these fields will be left undelimited since they must only be recognizable by the server host which hopefully can recognize its own job-ids and file-names.

SYNTAX

The following command descriptions are given in a BNF syntax. Names within angle brackets are non-terminal syntactic elements which are expanded in succeeding syntactic equations. Each equation has the

defined name on the left of the ::= and a set of alternative definitions, separated by vertical lines "|", on the right.

REINITIALIZE

REINIT

This command puts the user into a state identical to the state immediately after a successful connection to the RJE-server, prior to having sent any commands over the TELNET connection. The effective action taken is that of an ABORT and a flushing of all INPUT, OUTPUT and ID information. Naturally, the user is still responsible for any usage charges incurred prior to his REINIT command. The TELNET connection is not affected in any way.

USER

User = <user-id>

This command must be the first command over a new TELNET connection. As such, it initiates a "logon" sequence. The response to this command is one of the following:

1. User code in error.
2. Enter password (if user code ok).
3. Log-on ok, proceed (if no password requested).

Another USER command may be sent by the User at any time to change Users. Further input will then be charged to the new user. A server may refuse to honor a new user command if it is not able to process it in its current state (during input file transfer, for example), but the protocol permits the USER command at any time without altering previous activity. An incorrect subsequent USER command or its following PASS command are to be ignored with error response, leaving the original User logged-in.

It is permissable for a server to close the TELNET connection if the initial USER/PASS commands are not completed within a server specified time period. It is not required or implied that the "logged-on" User's user-id be the one used for file transfer or job execution, but only identifies the submitter of the command stream. Servers will establish their own rules relating user-id with the job-execution-user for Job or Output alteration commands.

Successful "log-on" always clears any previous Input or Output default parameters (INID, etc.).

PASS

Pass = <password>

This command immediately follows a USER command and completes the "log-on" procedure. Although a particular Server may not require a password and has already indicated "log-on ok" after the USER command, every Server must permit a PASS command (and possibly ignore it) and acknowledge it with a "log-on ok" if the log-on is completed.

BYE

BYE

This command terminates a USER and requests the RJE server to close the TELNET connection. If input transfer is not in progress, the TELNET connection may be closed immediately; if input is in progress, the connection should remain open for result response and then be closed. During the interim, a new USER command (and no other command) is acceptable.

An unexpected close on the TELNET connection will cause the server to take the effective action of an ABORT and a BYE.

INID/INPASS

INID = <user-id>
INPASS = <password>

The specified user-id and password will be sent in the File Transfer request to retrieve the input file. These parameters are not used by the Server in any other way. If this command does not appear, then the USER/PASS parameters are used.

INPATH/INPUT

INPATH = <file-id>
INPUT = <file-id>
INPUT

NOTE: The following syntax will be used for output as well.

<file-id>::= <host-socket> | <host-file>
<host-socket>::= <host>,<socket><attributes> |
<socket><attributes>
no <host> part implies the User-site host
<host>::= <integer>
<socket>::= <integer>

```
<integer>::= D<decimal-integer> | O<octal-integer> |
             H<hexadecimal-integer>
<host-file>::= <host><attributes>/<pathname>
<attributes>::= <empty> | :<transmission><code>
<transmission>::= <empty> | T | A | N
      <empty> implies default which is N for Input files
              and A for Output files
      T       specifies TELNET-like coding with embedded
              "crlf" for new-line, "ff" for new-page
      N       specifies FTP blocked transfer with record
              marks but without other carriage-control
      A       specifies FTP blocked records with ASA
              carriage-control
              (column 1 of image is forms control)
<code>::= <empty> | E
      <empty> specifies NVT ASCII code
      E specifies EBCDIC
<pathname>::= <any string recognized by the FTP Server at
              the site of the file>
```

The <file-id> syntax is the general RJE mechanism for specifying a particular file source or destination for input or output. If the <host-socket> form is used then direct transfer will be made by the RJE-Server to the named socket using the specified <attributes>. If the <host-file> form is used then the RJE-server will call upon its local FTP-user process to do the actual transfer. The data stream in this mode is either TELNET-like ASCII or blocked records (which may use column 1 for ASA carriage-control). Although A mode is permitted on input (column 1 is deleted) the usual mode is the default N. The output supplies carriage-control in the first character of each record ("blank" = single-space, "1" = new-page, etc.), while the optional N mode transfers the data only (as to a card punch, etc.).

The <pathname> is an arbitrary Unrecognized string which is saved by RJE-server and sent back over FTP to the FTP-server to retrieve or store the appropriate files.

INPATH or INPUT commands first store the specified <file-id> if one is supplied, and then the INPUT command initiates input. The INPATH name may be used to specify a file-id for later input and the INPUT command without file-id will cause input to initiate over a previously specified file-id. An INPUT "crlf" command with no previous <file-id> specified is illegal.

ABORT

ABORT

This command aborts any input retrieval in progress, discards already received records, and closes the retrieval connection. Note: ABORT with parameters is an Output Transmission control (see below).

OUTUSER/OUTPASS

OUTUSER = <user-id>
OUTPASS = <password>

The specified user-id and password will be sent in the File Transfer request to send the output file(s). These parameters are not used by the Server in any other way. If this command does not appear, then the USER/PASS parameters are used.

OUT

OUT <out-file> = <disp>

<out-file>::= <empty> | <job-file-id>
<empty> implies the primary print file of the job
<job-file-id>::= <string representing a specific output file from the job as recognized by the Server>
<disp>::= <empty><file-id> | (H) | (S)<file-id>|(D)
<empty> specifies Transmit then discard
(H) specifies Hold-only, do not transmit
(S) specifies Transmit and Save
(D) specifies discard without transmitting
Note: Parentheses are part of the above elements.

<file-id>::= (same as for INPUT command)

This command specifies the disposition of output file(s) produced by the job. Unspecified files will be Hold-only by default. The OUTUSER, OUTPASS, and OUT commands must be specified before the INPUT command to be effective. These commands will affect any following jobs submitted by this USER over this RJE-TELNET connection. A particular job may override these commands by NET control cards on the front of the input file.

Once output disposition is specified by this OUT command or by a NET OUT card, the information is kept with the job until final output disposition, and is modifiable by the CHANGE command.

On occasion, the server may find that the destination for the output is "busy" (i.e., RFC to either Server-FTP or specified socket is refused), or that the host which should receive the output is dead. In these cases, the server should wait several minutes and then try to transmit again.

OUTPUT RE-ROUTE

CHANGE <job-id><blank><out-file> = <disp>

This command changes the output disposition supplied with the job at submission. The <job-id> is assumed recognizable by the RJE-server, who may verify if this USER is authorized to modify the specified job. After the job is identified, the other information has the same syntax and semantics as the original OUT command. CHANGE command may be specified for a job-file-id which was not mentioned at submission time and has the same effect as an original OUT command.

OUTPUT CONTROLS DURING TRANSMISSION

<command><blank><count><blank><what>

<command>::= RESTART | RECOVER | BACK | SKIP |
ABORT | HOLD

These commands specify (respectively):
Restart the transmission (new RFC, etc.)
Recover restarts transmission from last FTP
Restart-marker-reply
(see FTP).
Back up the output "count" blocks
Skip the output forward "count" blocks
Abort the output, discarding it
Abort the output, but Hold it

<count>::= <empty> | <integer>
<empty> implies 1 where defined
<what>::= @<file-id> | <job-id><job-file-id>
<disp>::= (same as for OUT command)
<file-id>::= (same as for INPUT command)
<integer>::= (same as for INPUT command)
<job-id>::= <server recognized job identifier which was supplied
at INP completion by the server>

<job-file-id>::= <server recognized file identifier or if missing
then the prime printer output of the specified
job>

This collection of commands will modify the transmission of output in progress or recently aborted. If output transmission is cut-off before completion, then the RJE-server will either try to resend the entire file if the file's <disp> was Transmit-and-discard or will Hold the file for further User control if the <disp> was (S) transmit-and-Save. Either during transmission, during the Save part of a transmit-and-Save, or for a Hold-only file, the above commands may be used to control the transmission. The @<file-id> form of <what> is permitted only if transmission is actually in progress.

If the file's state is inconsistent with the command, then the command is illegal and ignored with reply.

STATUS

STATUS <job-id>
STATUS <job-id><blank><job-file-id>

These commands request the status of the RJE-server, a particular job, or the transmission of an output or input file, respectively. The information content of the Status reply is site dependent.

CANCEL/ALTER

CANCEL <job-id>
ALTER <job-id><blank><site dependent options>

These commands change the course of a submitted job. CANCEL specifies that the job is to be immediately terminated and any output discarded. ALTER provides for system dependent options such as changing job priority, process limits, Teminate without Cancel, etc.

OP

OP (any string)

The specified string is to be displayed to the Server site operator when any following job is initiated from the batch queue of the Server. This command usually appears in the input file as a NET OP control card, but may be a TELNET command. It is cancelled as an all-jobs command by an OP "crlf" command (no text supplied).

RJE CONTROL CARDS IN THE INPUT FILE

Certain RJE commands may be specified by control cards in the front of the input file. If these controls appear, they take precedence over the same command given thru the RJE-TELNET connection and affect only this specific job. All these RJE control cards must appear as the first records of the job's input-file. They all contain the control word NET in columns 1 through 3. Scanning for these controls stops when the first card without NET in col 1-3 is encountered.

The control commands appear in individual records and are terminated by the end-of-record (usually an 80 column card-image). Continuation is permitted onto the next record by the appearance of NET+ in columns 1-4 of the next record. Column 5 of the next record immediately follows the last character of the previous record.

```
NET OUTUSER = <user-id>
NET OUTPASS = <password>
NET OUT <out-file> = <disp>
NET OP <any string>
```

See the corresponding TELNET command for details. One option permitted by the NET OUTUSER and NET OUT controls not possible from the TELNET connection is specification of different OUTUSERs for different OUTS, since the TELNET stored and supplies only an initial OUTUSER, but the controls may change OUTUSERs before each OUT control is encountered.

RJE USE OF FILE TRANSFER PROTOCOL

Most non-TIP files will be transferred to or from the RJE-server through the FTP process. RJE-server will call upon its local FTP-user supplying the Host, File-pathname, User-id, Password, and Mode of the desired transfer. FTP-user will then connect to its FTP-server counterpart in the specified host and set up a transfer path. Data will then flow through the RJE-FTP interface in the Server, over the Network, from/to the foreign FTP-server and then from/to the specified File-pathname in the foreign host's file storage space. On output files, the file-pathname may be recognized by the foreign host as directions to a printer or the file may simply be stored; a User-RJE-process can supply an output <file-id> by default which is recognized by its own Server-FTP as routing to a printer.

Although many specifics of the RJE-Server/User-FTP interface are going to be site dependent, there are several FTP options which will be used in a standard way by RJE-Servers:

1. A new FTP connection will be initiated for each file to be transferred. The connection will be opened with the RJE User supplied User-id (OUTUSER or INUSER) and Password.

2. The data bytesize will be 8 bits.

3. The FTP Type, Structure, and Mode parameters are determined by the RJE transfer direction (I/O), and the <transmission> and <code> options supplied by the User:

I/O	<TRANS>	<CODE>	FTP-TYPE	FTP-STRUCTURE	FTP-MODE
I*	N	-	A	R	B
I	N	E	E	R	B
I	T	-	A	F	S
I	T	E	E	F	S
I	A	-	P	R	B
I	A	E	F	R	B
O*	A	-	P	R	B
O	A	E	F	R	B
O	N	-	A	R	B
O	N	E	E	R	B
O	T	-	A	F	S
O	T	E	E	F	S

(*indicates default)

4. The service commands used will be Retrieve for input and Append (with create) for output. The FTP pathname will be the <pathname> supplied by the RJE User.

5. On output in B form, the User-FTP at the RJE-Server site will send Restart-markers at periodic intervals (like every 100 lines, or so), and will remember the latest Restart-marker-reply with the file. If the file transfer is not completed and the <disp> is (S) then the file will be held pending User intervention. The User may then use the RECOVER command to cause a FTP restart at the last remembered Restart-marker-reply.

6. The FTP Abort command will be used for the RJE ABORT and CANCEL commands.

7. For transfers where the FTP-MODE is defined as B, the user FTP may optionally attempt to use H mode.

The specific form of the FTP commands used by an RJE-Server site, and the order in which they are used will not be specified in this protocol.

Errors encountered by FTP fall into three categories: a) access errors or no storage space error; b) command format errors; and c) transfer failure errors. Since the commands are created by the RJE-Server process, an error is a programming problem and should be logged for attention and the situation handled as safely as possible. Transmission failure or access failure on input cause an effective ABORT and user notification. Transmission failure on output causes RESTART or Save depending on <disp> (see OUT command). Access failure on output is a problem since the User may not be accessible. A status response should be queued for him, should he happen to inquire; a <disp> = (S) file should be Held; and a <disp> = <empty> transmit-and-discard file should be temporarily held and then discarded if not claimed. "Temporarily" is understood here to mean at least several days, since particularly in the case of jobs which generate voluminous output at great expense to the User, he should be given every chance to retrieve his rightful output. Servers may elect, however, to charge the User for the file-storage space occupied by the held output.

REPLIES OVER THE TELNET CONNECTION

Each action of the RJE-server, including entry of each TELNET command, is noted over the TELNET connection to the User. These RJE-server replies are formatted for Human or Process interpretation. They consist of a leading 3-digit numeric code followed by a blank followed by a text explanation of the message. The numeric codes are assigned by groups for future expansion to hopefully cover other protocols besides RJE (like FTP). The numeric code is designed for ease of interpretation by processes. The three digits of the code are interpreted as follows:

The first digit specified the "type" of response indicated:

000

These "replies" are purely informative, and are issued voluntarily by the Server to inform a User of some state of the server's system.

100

Replies to a specific status inquiry. These replies serve as both information and as acknowledgment of the status request.

200

Positive acknowledgment of some previous command/request. The reply 200 is a generalized "ok" for commands which require no other comment. Other 2xx replies are specified for specific successful actions.

300

Incomplete information supplied so far. No major problem, but activity cannot proceed with the input specified.

400

Unsuccessful reply. A request was correctly specified, but could not be correctly completed. Further attempts will require User commands.

500

Incorrect or illegal command. The command or its parameters were invalid or incomplete from a syntactic view, or the command is inconsistent with a previous command. The command in question has been totally ignored.

600-900

Reserved for expansion

The second digit specifies the general subject to which the response refers:

x00-x29

General purpose replies, not assignable to other subjects.

x30

Primary access. These replies refer to the attempt to "log-on" to a Server service (RJE, FTP, etc.).

x40

Secondary access. The primary Server is commenting on its ability to access a secondary service (RJE must log-on to a remote FTP service).

x50

FTP results.

x60

RJE results.

x70-x99

Reserved for expansion.

The final digit specifies a particular message type. Since the code is designed for an automaton process to interpret, it is not necessary for every variation of a reply to have a unique number, only that the basic meaning have a unique number. The text of a reply can explain the specific reason for the reply to a human User.

Each TELNET line (ended by "crlf") from the Server is intended to be a complete reply message. If it is necessary to continue the text of a reply onto following lines, then those continuation replies contain the special reply code of three blanks.

The assigned reply codes relating to RJE are:

000 General information message (time of day, etc.)
030 Server availability information
050 FTP commentary or user information
060 RJE or Batch system commentary or information
100 System status reply
150 File status reply
151 Directory listing reply
160 RJE system general status reply
161 RJE job status reply
200 Last command received ok
201 An ABORT has terminated activity, as requested
202 ABORT request ignored, no activity in progress
203 The requested Transmission Control has taken effect
204 A REINIT command has been executed, as requested
230 Log-on completed
231 Log-off completed, goodbye.
232 Log-off noted, will complete when transfer done
240 File transfer has started
250 FTP File transfer started ok
251 FTP Restart-marker-reply
   Text is: MARK yyyy = mmmm
      where yyyy is data stream marker value (yours)
      and mmmm is receiver's equivalent mark (mine)
252 FTP transfer completed ok
253 Rename completed
254 Delete completed
260 Job <job-id> accepted for processing
261 Job <job-id> completed, awaiting output transfer
262 Job <job-id> Cancelled as requested
263 Job <job-id> Altered as requested to state <status>
264 Job <job-id>,<job-file-id> transmission in progress
300 Connection greeting message, awaiting input
301 Current command not completed (may be sent after
   suitable delay, if not "crlf")
330 Enter password (may be sent with hide-your-input mode)
360 INPUT has never specified an INPATH
400 This service is not implemented
401 This service is not accepting log-on now, goodbye.
430 Log-on time or tries exceeded, goodbye.
431 Log-on unsuccessful, user and/or password invalid
432 User not valid for this service
434 Log-out forced by operator action, please phone site
435 Log-out forced by system problem
436 Service shutting down, goodbye
440 RJE could not log-on to remote FTP for input transfer
441 RJE could not access the specified input file thru FTP
442 RJE could not establish <host-socket> input connection

443 RJE could not log-on to remote FTP for output delivery
444 RJE could not access file space given for output
445 RJE could not establish <host-socket> output connection
450 FTP: The named file does not exist (or access denied)
451 FTP: The named file space not accessable by YOU
452 FTP: Transfer not completed, data connection closed
453 FTP: Transfer not completed, insufficient storage space
460 Job input not completed, ABORT performed
461 Job format not acceptable for processing, Cancelled
462 Job previously accepted has mysteriously been lost
463 Job previously accepted did not complete
464 Job-id referenced by STATUS, CANCEL, ALTER, CHANGE, or Transmission Control is not known (or access denied)
465 Request Alteration is not permitted for the specified job
466 Un-deliverable, un-claimed output for <job-id> discarded
467 Requested REINIT not accomplished
500 Last command line completely unrecognized
501 Syntax of the last command is incorrect
502 Last command incomplete, parameters missing
503 Last command invalid, illegal parameter combination
504 Last command invalid, action not possible at this time
505 Last command conflicts illegally with previous command(s)
506 Requested action not implemented by this Server
507 Job <job-id> last command line completely unrecognized
508 Job <job-id> syntax of the last command is incorrect
509 Job <job-id> last command incomplete, parameters missing
510 Job <job-id> last command invalid, illegal parameter combination
511 Job <job-id> last command invalid, action impossible at this time
512 Job <job-id> last command conflicts illegally with previous command(s)

SEQUENCING OF COMMANDS AND REPLIES

The communication between the User and Server is intended to be an alternating dialogue. As such, the User issues an RJE command and the Server responds with a prompt primary reply. The User should wait for this initial success or failure response before sending further commands.

A second type of reply is sent by Server asynchronously with respect to User commands. These replies report on the progress of a job submission caused by the INPUT command and as such are secondary replies to that command.

The final class of Server "replies" are strictly informational and may arrive at any time. These "replies" are listed below as spontaneous.

COMMAND-REPLY CORRESPONDENCE TABLE

COMMAND	SUCCESS	FAILURE
REINIT	204	467,500-505
USER	230,330	430-432,500-505
PASS	230	430-432,500-505
BYE	231,232	500-505
INID	200	500-505
INPASS	200	500-505
INPATH	200	500-505
INPUT	240	360,440-442,500-505
sec. input retrieval	260	460,461
sec. job execution	261	462,463
sec. output transmission -		443-445,466
ABORT (input)	201,202	500-505
OUTUSER	200	500-505
OUTPASS	200	500-505
OUT	200	500-505
CHANGE	200	500-505
RESTART/RECOVER/BACK /SKIP/ABORT (output)/HOLD	203	464,500-506
STATUS	1xx,264	460-465,500-505
CANCEL	262	464,500-506
ALTER	263	464,465,500-506
OP	200	500-505
Spontaneous	0xx,300,301	434-436

Note: For commands appearing on cards, a separate set of error codes is provided (507-512). Since these error replies are "asynchronously" sent, and thus could cause some confusion if the user is in the process of submitting a new job after the present one, the error replies must identify which job has the faulty card(s).

TYPICAL RJE SCENARIOS

TIP USER WANTING HOT CARD READER TO HOSTX

1. TIP user opens TELNET connection to HOSTX socket 5

2. Commands sent over TELNET to RJE

   ```
   USER=myself
   PASS=dorwssap
   OUT=H70002
   INPUT=H50003
   ```

3. RJE-server connects to the TIP's device 5 and begins reading. When end-of-job card is recognized, the job is queued to run. The connection to the card reader is still open for more input as another job.

4. The first job finishes. A connection to the TIP's device 7 is established by RJE-server and the output is sent as an NVT stream.

5. Continue at any time with another deck at step 3.

TIP WITH JOB-AT-A-TIME CARD READER

1. thru 4) the same but User closes Reader after the deck

2. The output finishes and the printer connection closes.

3. INPUT may be typed any time after step 3 finishes and another job will be entered starting at 3.

HOSTA USER RUNS JOB AT HOSTC, INPUT FROM HOSTB

1. User TELNET connects to HOSTC socket 5 for RJE

   ```
   USER=roundabout
   PASS=aaabbbc
   OUTUSER=roundabl
   OUT=:E/.sysprinter
   OUT puncher = (S)HOSTB:NE/my.savepunch
   INUSER=rounder
   INPASS=x.x.x
   INPUT=HOSTB:E/my.jobinput
   ```

2. The RJE-server has FTP retrieve the input from HOSTB using User-id of "rounder" and Password of "x.x.x" for file named "my.jobinput".

3. The job finishes. RJE-server uses FTP to send two files: the print output is sent to HOSTA in EBCDIC with ASA carriage control to file ".sysprinter" while the file known as "puncher" is sent to HOSTB in EBCDIC without carriage-control to file "my.savepunch".

4. when the outputs finish, RJE-server at HOSTC discards the print file but retains the "puncher" file.

5. The User who has signed out after job submission has gotten his output and checked his file "my.savepunch" at HOSTB. He deletes the saved copy at HOSTC by re-calling RJE at HOSTC.

   ```
   USER=roundabout
   PASS=aaabbbcc
   ABORT job 123 puncher
         or
   CHANGE job 123 puncher = (D)
   ```